

SYSTEM AND METHOD FOR FILE MANAGEMENT

Patent Number: JP10011345
Publication date: 1998-01-16
Inventor(s): INOUE SAKAE
Applicant(s): TOSHIBA CORP
Requested Patent: ☒ JP10011345
Application Number: JP19960166048 19960626
Priority Number(s):
IPC Classification: G06F12/00 ; G06F12/00
EC Classification:
Equivalents:

Abstract

PROBLEM TO BE SOLVED: To provide a system and method for file management which can repair hierarchical relation, whose consistency is lost owing to trouble, under simple control.

SOLUTION: Unique IDs 48 which are enumerated uniquely in a tree structure are given to (i) nodes 47 held corresponding to the respective nodes constituting the tree structure, and when an (i) node 47 is updated, the unique ID 48 is also newly enumerated. Each time the (i) code 47 is updated, maintenance data 42 including the update state and the latest enumerated unique ID are generated and held in a volume maintenance space 41. If trouble occurs during the update of the (i) node 47, the latest maintenance data 42 are retrieved from the maintenance data 42 showing the completion of update states and the (i) node 47 including the enumerated unit ID 47 newer than the unique ID 43 included in the retrieved maintenance data 42.

Data supplied from the esp@cenet database - I2

9/27/01

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-11345

(43) 公開日 平成10年(1998) 1月16日

(51) Int.Cl. ⁹	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/00	5 3 1		G 0 6 F 12/00	5 3 1 R
	5 2 0			5 2 0 J

審査請求 未請求 請求項の数 7 O L (全 8 頁)

(21) 出願番号 特願平8-166048

(22) 出願日 平成8年(1996) 6月26日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 井上 栄

東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内

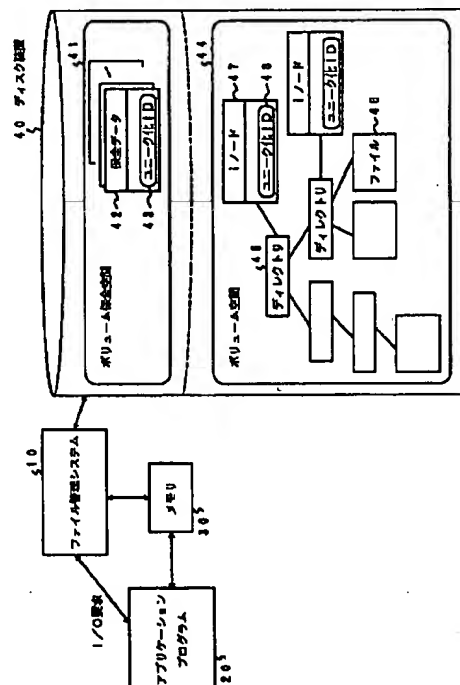
(74) 代理人 弁理士 鈴江 武彦 (外6名)

(54) 【発明の名称】 ファイル管理システムおよびファイル管理方法

(57) 【要約】

【課題】 障害によって整合性の損なわれた階層関係を簡単な制御で修復することのできるファイル管理システムおよびファイル管理方法を提供する。

【解決手段】 木構造を構成する各節点に対応して保持されるiノード47に、この木構造内で一意に採番されるユニーク化ID48を加え、このiノード47が更新されるときには、そのユニーク化ID48も新たに採番していく。一方、このiノード47の更新が行なわれるごとに、その更新状況とそのとき採番された最新のユニーク化IDを含む保全データ42を生成してボリューム保全空間41に保持しておく。そして、iノード47の更新中に障害が発生したときに、更新状況が完了を示す保全データ42の中から最新の保全データ42を検索し、その検索された保全データ42に含まれるユニーク化ID43よりも新しく採番されたユニーク化ID47を含むiノード47を修復する。



【特許請求の範囲】

【請求項1】 各節点にファイルやディレクトリなどのデータを蓄える木構造のファイル管理システムであって、前記各節点の階層的な関係を含むノード情報を前記各節点それぞれに対応して保持するファイル管理システムにおいて、

前記ノード情報は、前記木構造内で一意に採番される識別番号をその構成要素に含んでなり、

前記データの更新に伴って前記ノード情報を更新するときに、その対象となる一連のノード情報群に含まれる識別番号を新たに採番する採番手段と、

前記一連のノード情報群の更新状況と前記採番手段により採番された最新の識別番号とを含む前記データの更新ごとに生成される保全情報を記憶する保全情報記憶手段と、

前記ノード情報の更新中に障害が発生したときに、前記更新状況が完了を示す保全情報の中から最新の保全情報を前記保全情報記憶手段より検索し、その検索された保全情報に含まれる識別番号と前記ノード情報に含まれる識別番号とから前記ノード情報を修復する修復手段とを具備してなることを特徴とするファイル管理システム。

【請求項2】 前記採番手段は、予め与えられた初期値をインクリメントしていくことによって一意な識別番号を採番し、

前記修復手段は、前記検索された保全情報に含まれる識別番号よりも大きな値の識別番号を含むノード情報を修復対象とする請求項1記載のファイル管理システム。

【請求項3】 前記データおよびノード情報が追記型の記憶媒体に記憶されるファイル管理システムであって、前記修復手段は、前記更新後のノード情報を削除して、その更新により不定状態となっている更新前のノード情報を復帰させる手段を具備してなることを特徴とする請求項1または2記載のファイル管理システム。

【請求項4】 前記ノード情報と保全情報とを同一記憶装置内の論理的に独立して確保された領域に別々に記憶することを特徴とする請求項1、2または3記載のファイル管理システム。

【請求項5】 複数の記憶装置それぞれが木構造を形成するファイル管理システムであって、前記採番手段は、前記記憶装置それぞれで一意となる識別番号を採番することを特徴とする請求項1、2、3または4記載のファイル管理システム。

【請求項6】 複数の記憶装置それぞれが木構造を形成するファイル管理システムであって、前記採番手段は、前記システム内で一意となる識別番号を採番することを特徴とする請求項1、2、3または4記載のファイル管理システム。

【請求項7】 各節点にファイルやディレクトリなどのデータを蓄える木構造のファイル管理方法であって、前記各節点の階層的な関係を含むノード情報を前記各節点

それぞれに対応して保持するファイル管理方法において、

前記ノード情報は、前記木構造内で一意に採番される識別番号をその構成要素に含んでなり、

前記データの更新に伴って前記ノード情報を更新するときに、その対象となる一連のノード情報群に含まれる識別番号を新たに採番し、

前記一連のノード情報群の更新状況と前記採番された最新の識別番号とを含む前記データの更新ごとに生成される保全情報を記憶しておき、

前記ノード情報の更新中に障害が発生したときに、前記更新状況が完了を示す保全情報の中から最新の保全情報を前記保全情報記憶手段より検索し、その検索された保全情報に含まれる識別番号と前記ノード情報に含まれる識別番号とから前記ノード情報を修復することを特徴とするファイル管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、木構造でファイルやディレクトリなどのデータを管理するファイル管理システムおよびファイル管理方法に係り、特に障害によって整合性の損なわれた階層関係を簡単な制御で修復することのできるファイル管理システムおよびファイル管理方法に関する。

【0002】

【従来の技術】近年、様々な業種でコンピュータシステムが導入されており、いまやコンピュータシステムは、業務遂行上必要不可欠なものとなりつつある。また、この導入に伴って、取り扱うデータ量が膨大となり、その種類も多様化の一途を辿っている。このような状況においては、所望のデータをいかに効率良く探索することができるかが、コンピュータシステムの性能評価を大きく左右する。そして、このデータ探索をより効率良く行なうため、今日までに種々のデータ構造の概念が生み出されてきており、その中の一つとして木構造が存在する。

【0003】木構造は、データの階層関係により木の形状を概念的に構成するといった構造であり、その各節点にファイルやディレクトリなどのデータを蓄える構造となっている。たとえば、オペレーティングシステムにUNIXを採用するコンピュータ（以下、UNIXコンピュータという。）で構築されるファイルシステムでは、いくつかのファイルをディレクトリによって区分けして管理しており、このディレクトリには、配下のファイル（これがさらにディレクトリであってもよい）を指し示すアドレスポインタが含まれている。すなわち、このディレクトリおよびファイルの階層構造が、前述した木構造を成している。

【0004】UNIXコンピュータのファイルシステムでは、この配下のファイルやディレクトリを指し示すも

のとして、ディスクインデックスノード(disk index node:以下、iノードという。)を定義する。このiノードは、モード、所有者名、タイムスタンプ、大きさ、ブロック数、参照カウント、直接ブロックおよび間接ブロックから構成される。

【0005】図7には、UNIXコンピュータで構築されるファイルシステムの木構造の一例が示されている。ここでは、この図7に示したファイルシステムが、追記型の記憶媒体上に構築されている場合を考える。この場合に、たとえばファイル名が「file」であるファイルに新たにデータを追加するといった更新を実施すると、この「file」の実体であるデータの格納領域の移動が発生するため、この「file」の格納領域を指し示すアドレス(X5)を更新する必要がある。したがって、このアドレス(X5)を保持するbin用のiノードも更新する必要がある。さらにこのbin用iノードの更新は、bin用iノードの格納領域の移動を意味するため、このbin用iノードの格納領域を指し示すアドレス(X3)を保持するルートディレクトリ用のiノードを更新しなければならない。すなわち、このような追記型の記憶媒体上に木構造のファイルシステムを構築した場合には、配下のファイルの更新に伴って、階層的に関連のもつiノードをすべて更新していかなければならない。

【0006】従来のファイルシステムでは、更新の対象となるファイルやディレクトリなどのデータを一つのiノードで管理するように構成されている。したがって、前述したような一連のiノードの更新中に障害が発生したような場合には、iノードが破壊されてファイルシステムに異常をきたすという問題があった。

【0007】また、従来のファイルシステムでは、ディレクトリから各iノードを指し示すように構成されていたため、ディレクトリの更新中に障害が発生したような場合であっても、ディレクトリが破壊されてiノードを参照することができなくなる結果、同様にファイルシステムに異常をきたすという問題があった。

【0008】さらに、従来のファイルシステムでは、このような原因でファイルシステムに異常をきたした場合には、もとの状態にファイルシステムを修復することができないという問題があった。

【0009】

【発明が解決しようとする課題】このように、従来の木構造のファイルシステムでは、iノードによりその階層的な関係を管理していたために、このiノードの更新中に障害が発生したような場合には、階層関係の整合性が損なわれてしまい、ファイルシステムに異常をきたしてしまうといった問題があった。

【0010】また、このiノードは、ディレクトリに含まれるポインタによって参照されるようになっており、ディレクトリが破壊された場合には、iノードを参照す

ることができなくなってしまうため、同様にファイルシステムに異常をきたしてしまうといった問題があった。

【0011】さらに、従来のファイルシステムでは、このような原因でファイルシステムに異常をきたした場合に、もとの状態にファイルシステムを修復することができないという問題があった。

【0012】この発明は、このような実情に鑑みてなされたものであり、障害によって整合性の損なわれた階層関係を簡単な制御で修復することのできるファイル管理システムおよびファイル管理方法を提供することを目的とする。

【0013】

【発明を解決するための手段】この発明は、各節点にファイルやディレクトリなどのデータを蓄える木構造のファイル管理システムであって、前記各節点の階層的な関係を含むノード情報を前記各節点それぞれに対応して保持するファイル管理システムにおいて、前記ノード情報は、前記木構造内で一意に採番される識別番号をその構成要素に含んでなり、前記データの更新に伴って前記ノード情報を更新するときに、その対象となる一連のノード情報群に含まれる識別番号を新たに採番する採番手段と、前記一連のノード情報群の更新状況と前記採番手段により採番された最新の識別番号とを含む前記データの更新ごとに生成される保全情報を記憶する保全情報記憶手段と、前記ノード情報の更新中に障害が発生したときに、前記更新状況が完了を示す保全情報の中から最新の保全情報を前記保全情報記憶手段より検索し、その検索された保全情報に含まれる識別番号と前記ノード情報に含まれる識別番号とから前記ノード情報を修復する修復手段とを具備してなることを特徴とする。

【0014】この発明においては、たとえばデータの更新に伴って、関連する一連のノード情報を更新するときに、このノード情報群に対応した保全情報を生成して記憶しておく。この発明では、ノード情報として、木構造内で一意に採番される識別番号を含んでいる。この識別番号は、採番手段が、たとえば予め与えられた初期値をインクリメントしていくことによって一意な番号を採番する。そして、ノード情報を更新する際は、そのノード情報に含まれる識別番号を新たに採番して書き替える。

【0015】一方、この一連のノード情報群の更新に伴って生成される保全情報には、その更新の状況と、この更新によって採番された最新の識別番号とが含まれる。採番手段がインクリメントして番号を採番している場合には、最も大きい番号の識別番号が最新の識別番号ということになる。

【0016】ここで、ノード情報の更新中に障害が発生した場合を考える。ノード情報の更新中に障害が発生すると、この発明においては、最新の保全情報に含まれる更新状況が更新中を示していることとなり、これによ

て修復作業の必要性を知得することができることになる。そして、この修復作業は、修復手段によって実施され、修復手段は、まず更新状況が完了を示す保全情報の中から最新の保全情報を検索する。このとき検索された保全情報には、整合性が保証される障害時点に最も近い点の状態が保持されており、この保全情報に含まれる識別番号を得ることによって、修復対象とするノード情報を特定する。この場合には、採番手段がインクリメントして番号を採番しているため、ノード情報に含まれる識別番号が、この検索された保全情報に含まれる識別番号よりも大きな値であるノード情報を修復対象とすればよいことになる。

【0017】たとえば、追記型の記憶媒体では、データを更新するときに、更新前のデータ、すなわち更新対象とするデータの格納領域を不定状態にし、更新後のデータを新たに確保した領域に格納する。したがって、修復手段は、検索された保全情報に含まれる識別番号よりも大きな値の識別番号を含むノード情報を削除し（不定状態にし）、このノード情報が格納された際に不定状態とされた更新前のノード情報をアクセス可能に復帰させる。

【0018】すなわち、この発明においては、更新前のデータをアーカイブとして利用することになり、ファイルシステムの保全性を飛躍的に向上させるとともに、資源の有効利用も図られることになる。

【0019】

【発明の実施の形態】以下、図面を参照してこの発明の実施の形態を説明する。図1には、この発明の実施形態に係るコンピュータシステムの概略構成が示されている。本実施形態のファイル管理システム10は、アプリケーションプログラム（ユーティリティプログラムを含む）20から、ディスク装置40に対する入力要求を受け取ると、その要求されたデータをディスク装置40から読み出して、メモリ30に格納する。一方、ディスク装置40に対する出力要求を受け取った場合には、その要求されたデータをメモリ30から読み出して、ディスク装置40に格納する。

【0020】このディスク装置40は、追記型の記憶媒体であり、また、ボリューム保全空間41とボリューム空間44とに論理的に分割されている。したがって、ボリューム空間44に格納されたデータに対して論理的に不正な入出力が行なわれた場合であっても、その影響はボリューム保全空間41には及ばない。

【0021】このボリューム保全空間41には、更新処理が発生することによって生成される保全データ42が保持される。この保全データ42には、図2に示したように、後述するiノードを一意にするための最新のユニーク化IDと、ボリューム空間44がオープンされている状態か、またはクローズされている状態を示す保全状態と、この保全データ42にアクセスしたときの時刻を示

すタイムスタンプとが含まれている。

【0022】また、ボリューム空間44には、管理対象となる実際のデータ群が木構造で保持される。この木構造は、各節点にディレクトリ45およびファイル46などを蓄えており、その階層的な関係を示す情報として、ディレクトリ45またはファイル46の少なくとも一方から構成される各節点に対応させてiノード47が設けられる。このiノード47には、図3に示したように、モードや所有者名などといった従前の情報に加えて、ユニーク化IDが含まれている。このユニーク化IDを各iノード47の構成要素として加えたことが本実施形態の特徴であり、この取扱いについて以下に説明する。

【0023】図4には、ディスク装置40上に構築されるファイルシステムの木構造の一例が示されている。図4に示したように、このファイルシステムは、ルートディレクトリを木の根とし、このルートディレクトリは、配下にuserおよびbinの2つのサブディレクトリを備えている。また、このbinは、配下にcatおよびfileの2つのファイルを備えている。そして、このディレクトリまたはファイルによって構成される各接点それぞれに、その階層的な関係を示したiノードが備えられる。したがって、ファイル管理システム10は、このiノードを辿ることによって所望のデータを探索する。

【0024】いま、各iノードに含まれるユニーク化IDには、図4に示したような値がセットされているものとする。すなわち、たとえばルートディレクトリ用iノードのユニーク化IDには“1”、bin用iノードのユニーク化IDには“4”、およびfile用iノードのユニーク化IDには“7”がそれぞれセットされているものとする。このユニーク化IDは、予め設定された初期値から（ここでは“1”）昇順に採番されるものであり、新たなiノードが生成されたときにインクリメントされるが、既存のiノードが更新されたときにもインクリメントされた新たなユニーク化IDが採番される。

【0025】なお、この図4に示した状態では、ボリューム保全空間41に保持される最新の保全データ42には、ユニーク化IDとして“7”、保全状態として“クローズ”がそれぞれセットされていることになる。

【0026】ここで、サブディレクトリbinの配下にファイルfile1を追加することを考える。この場合には、図5に示したように、file1が格納される領域を指し示すためのiノード（file1用iノード）が生成される。そして、この生成されたfile1用iノードを指し示すアドレス（X13）を保持するために、bin用iノードの更新、すなわちbin用iノードの格納領域の移動が発生する。さらに、このbin用iノードの格納領域を指し示す新たなアドレス（X12）を保持するルートディレクトリ用のiノードの更新が発生する。

【0027】このとき、ファイル管理システム10は、この一連の更新に対して、一つの保全データ42を生成する。そして、その更新処理の開始時に、保全状態を“オープン”とし、一方、更新処理の完了時には、最新（すなわち最大値）のユニーク化IDをセットするとともに、保全状態を“クローズ”とする。本実施形態のディスク装置40のように、追記型の記憶媒体の場合においては、データを更新するときに、更新前のデータ、すなわち更新対象とするデータの格納領域を不定状態にし、更新後のデータを新たに確保した領域に格納する。したがって、更新前のiノードなどは依然としてボリューム空間44に保持されている状態であり、本実施形態のファイル管理システム10では、この更新前のデータをアーカイブとして利用する。

【0028】ここで、このiノードの更新中に障害が発生した場合を考える。この場合、ボリューム保全空間41に保持される最新の保全データ42の保全状態は、“オープン”のみであり、これにより、ファイル管理システム10は、いずれかのiノード47に不整合が生じている状態であることを知得する。

【0029】このとき、ファイル管理システム10は、ボリューム保全空間41に保持される保全データ42の中から、保全状態が“クローズ”であって、タイムスタンプの示す時刻が最も新しい保全データ42を検索する。そして、この検索した保全データ42からユニーク化IDを取り出して、ボリューム空間44に保持されるiノード47の中から、検索した保全データ42のユニーク化IDよりも大きな値のユニーク化IDを含むiノード47を検索する。そして、この検索されたiノード47を修復対象として、そのiノード47を削除するとともに、削除したiノード47のアーカイブとなっているiノード47を復帰させる。

【0030】これにより、iノード47の更新中に障害が発生した場合であっても、損なわれたiノード47の整合性を修復できることになり、ディスク装置40の保全性を向上させることが可能となる。

【0031】ここで、図6を参照してファイル管理システム10の整合性確保の動作原理を説明する。ファイル管理システム10は、ボリューム保全空間41に保全データ42が存在するかどうか判定し（ステップS1）、保全データ42が存在しない場合には（ステップS1のN）、ディスク装置40が初期化状態にあると認識する（ステップS2）。一方、保全データ42が存在した場合（ステップS1のY）、ファイル管理システム10は、その中から最新の保全データ42を検索し（ステップS3）、この検索された保全データ42に含まれる保全状態が“オープン”か“クローズ”かを参照する（ステップS4）。

【0032】この最新の保全データ42に含まれる保全状態が“クローズ”であった場合（ステップS4の

N）、ファイル管理システム10は、ディスク装置40の整合性は完全に保たれていると認識する（ステップS5）。一方、保全状態が“オープン”であった場合には（ステップS4のY）、ファイル管理システム10は、ディスク装置40の整合性は損なわれていると認識する。そして、ファイル管理システム10は、ボリューム保全空間41に保持される保全データ42の中から、保全状態が“クローズ”であって、タイムスタンプの示す時刻が最も新しい保全データ42を検索し、その検索した保全データ42に含まれるユニーク化IDを取得する（ステップS6）。

【0033】そして、ファイル管理システム10は、この取得したユニーク化IDよりも大きな値のユニーク化IDを含むiノード47をボリューム空間44に保持されるiノード47の中から検索し、その検索されたiノード47を削除してアーカイブとなっているiノード47を復帰させる（ステップS7）。

【0034】これにより、ファイル管理システム10は、ディスク装置40の整合性を効率良く確保することができる。なお、本実施形態では記憶媒体として追記型の記憶媒体を例に説明したが、これに限定されるものではなく、たとえば書き換え型の記憶媒体の場合であっても本発明は適用可能である。

【0035】

【発明の効果】以上詳述したように、この発明によれば、更新時に新たに採番される一意の識別番号をノード情報の構成要素とし、かつ最新の識別番号とシステムの保全状態とを保全情報として別途保持しているため、ノード情報の更新中に障害が発生したような場合であっても、保全情報に含まれる識別番号とノード情報に含まれる識別番号とから、損なわれた階層的な関係を修復することが可能となるため、システム全体の保全性を飛躍的に向上させることが可能となる。また、更新前の情報をアーカイブとして利用するために、新たな記憶領域などを確保する必要がなく、資源の有効利用を図ることが可能となる。

【図面の簡単な説明】

【図1】この発明の実施形態に係るコンピュータシステムの概略構成を示す図。

【図2】同実施形態の保全データの構成を示す図。

【図3】同実施形態のディスクインデックスノード（iノード）の構成を示す図。

【図4】同実施形態のディスク装置上に構築されるファイルシステムの本構造の一例を示す図。

【図5】同実施形態のファイルシステムの更新動作を説明するための図。

【図6】同実施形態のファイル管理システムの整合性確保の動作原理を説明するためのフローチャート。

【図7】オペレーティングシステムにUNIXを採用するコンピュータで構築されるファイルシステムの本構造

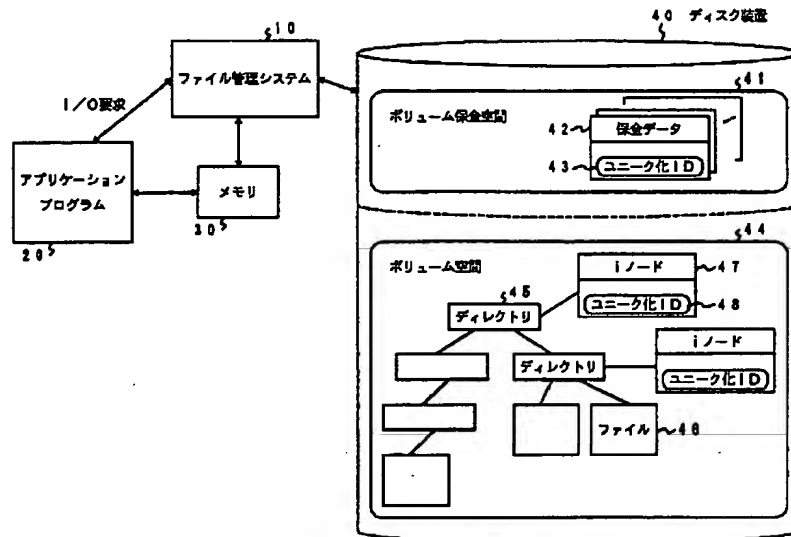
の一例を示す図。

【符号の説明】

10…ファイル管理システム
20…アプリケーションプログラム
30…メモリ
40…ディスク装置
41…ボリューム保全空間

42…保全データ
43…ユニーク化ID
44…ボリューム空間
45…ディレクトリ
46…ファイル
47…ディスクインデックスノード(iノード)
48…ユニーク化ID

【図1】



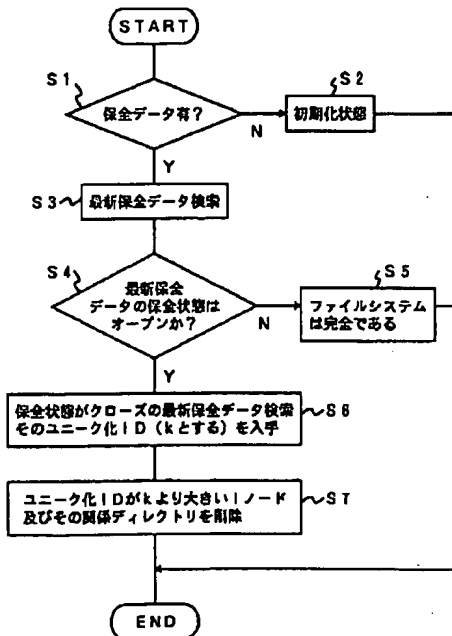
【図3】

モード
所有者名
タイムスタンプ
大きさ
ブロック数
参照カウント
ユニーク化ID
直接ブロック
間接ブロック

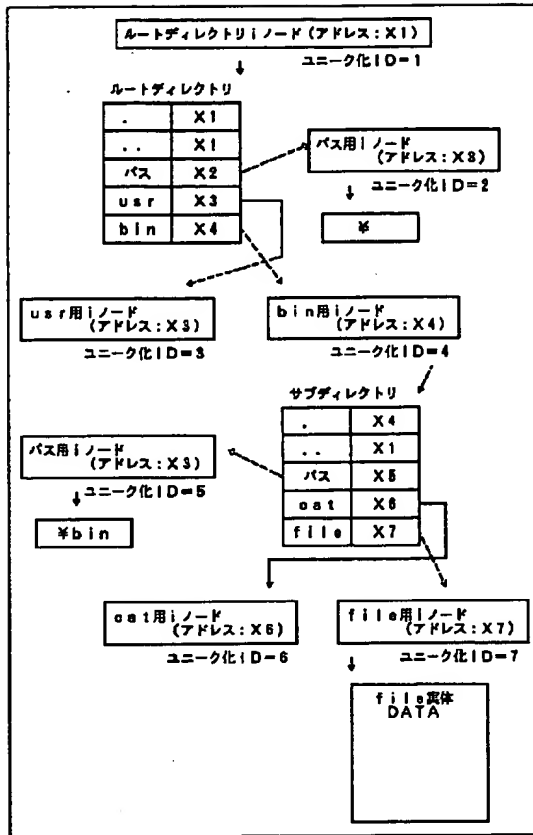
【図2】

ユニーク化ID
保全状態 （オープン：ボリューム空間がオープン） （クローズ：ボリューム空間がクローズ）
タイムスタンプ

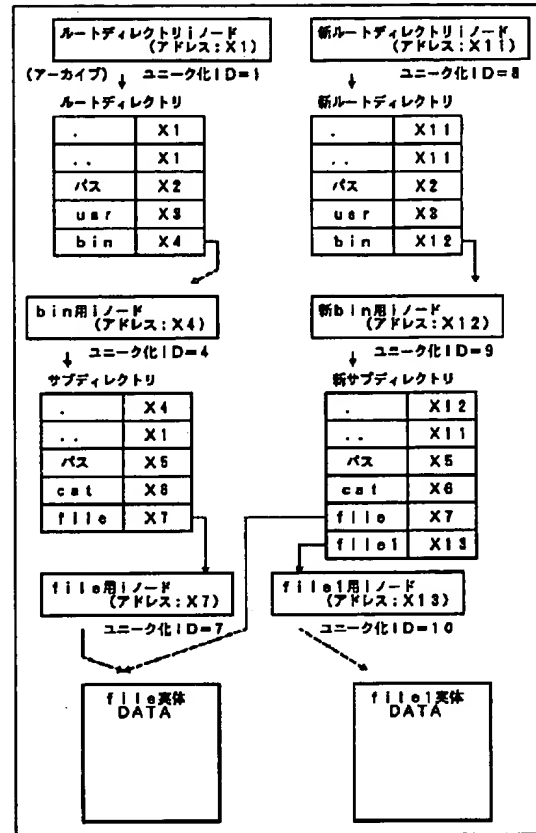
【図6】



【図4】



【図5】



【図7】

